

VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Absolvování individuální odborné praxe

Individual Professional Practice in the Company

Zadání bakalářské práce

Student: **Tomáš Ptáček**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: **Absolvování individuální odborné praxe**
Individual Professional Practice in the Company

Jazyk vypracování: čeština

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: ECHOpix s.r.o.
2. Struktura závěrečné zprávy:
 - a) Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta.
 - b) Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti.
 - c) Zvolený postup řešení zadaných úkolů.
 - d) Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe.
 - e) Znalosti či dovednosti scházející studentovi v průběhu odborné praxe.
 - f) Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení.

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vede odbornou praxi studenta.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. David Ježek, Ph.D.**

Konzultant bakalářské práce: Ing. Jan Slováček

Datum zadání: 01.09.2015

Datum odevzdání: 29.04.2016



doc. Dr. Ing. Eduard Sojka
vedoucí katedry





prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární
prameny a publikace, ze kterých jsem čerpal.

V Ostravě 27. dubna 2016


.....

Souhlasím se zveřejněním této bakalářské práce dle požadavků čl. 26, odst. 9 Studijního a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava.

V Ostravě 27. dubna 2016



Rád bych poděkoval svému vedoucímu práce, panu Ing. Davidu Ježkovi, Ph.D. za konzultace, rady a tipy v průběhu psaní této práce. Dále také mému kolegovi ve firmě, Marku Kajfoszovi, za trpělivost a spolupráci při řešení problémů a v neposlední řadě také společnosti ECHOpix s.r.o. za příležitost pro ně bakalářskou práci formou praxe vykonat.

Abstrakt

Obsahem této bakalářské práce je popsat mé působení ve firmě ECHOpix s.r.o. na pozici developera mobilních aplikací pro mobilní zařízení s operačním systémem Android. V první části se zmíním o zaměření a oblast působení této firmy, poté popíšu jednotlivá zadání, případné problémy a jejich řešení a v neposlední řadě také využívané technologie. Na závěr práce bych rád shrnul jaké znalosti jsem při absolvování této praxe nabyl a pokusím se je dát do kontrastu se znalostmi, které jsem získal ve škole.

Klíčová slova: Bakalářská práce, odborná praxe, ECHOpix s.r.o., Java, SQL, MySQL, Android, Android framework

Abstract

Content of this bachelor thesis is my work experience in ECHOpix s.r.o. as developer of mobile applications with the focus on mobile devices running Android operation system. In the first part I will talk about the area of expertise of said company, afterwards I will talk about my tasks, problems, solutions associated with these tasks and technologies I used while solving them. Finally I will mention what experiences I have gained during my work and I would like to put them in contrast with the knowledge I've gained during my studies.

Key Words: Bachelor thesis, professional experience, ECHOpix s.r.o., Java, SQL, MySQL, Android, Android framework

Obsah

Seznam použitých zkratek a symbolů	8
Seznam obrázků	9
Seznam výpisů zdrojového kódu	10
1 Úvod	11
2 Základní informace o firmě	12
2.1 Odborné zaměření firmy	12
2.2 Mé pracovní zařazení	12
3 Detailní přehled jednotlivých úkolů	13
3.1 Společné technologie a nástroje	13
3.2 Aplikace Oskárek	14
3.3 Aplikace Dohodnuto	19
4 Závěr	25
4.1 Časová náročnost projektů	25
4.2 Dosavadní znalosti	25
4.3 Nabyté znalosti	25
4.4 Celkové zhodnocení	26
Literatura	27

Seznam použitých zkratk a symbolů

NFC	– Near Field Communication
NDEF	– NFC Data Exchange Format
ORM	– Object-relational mapping
API	– Application Programming Interface
MIME	– Multipurpose Internet Mail Extensions
URI	– Uniform Resource Identifier
IDE	– Integrated Development Environment
REST	– Representational State Transfer
SOAP	– Simple Object Access Protocol
PDF	– Portable Document Format
AWS	– Amazon Web Services
EC2	– Elastic Compute Cloud
VPS	– Virtual Private Server
VPC	– Virtual Private Cloud
PHP	– PHP: Hypertext Preprocessor
JSON	– JavaScript Object Notation
POJO	– Plain Old Java Object
HTTP	– Hypertext Transfer Protocol
CRUD	– Create, Read, Update, Delete
HW	– Hardware
SW	– Software
GUI	– Graphical User Interface

Seznam obrázků

1	Popularita PHP frameworků [11]	14
2	Ukázka aplikace Oskárek	15
3	Ukázka zhodnocení událostí	18
4	Struktura projektu Dohodnuto	20
5	Pseudo struktura databáze	22
6	Dohodnuto - Android aplikace	23
7	Dohodnuto - Webová aplikace	24

Seznam výpisů zdrojového kódu

1	Ukázka kódu zápisu na NFC Tag	17
---	---	----

1 Úvod

Pro svou bakalářskou práci jsem zvolil možnost využít odborné praxe. Své rozhodnutí jsem učinil na základě několika věcí. V první řadě jsem chtěl nabrat zkušenosti z reálného pracoviště, kde bych se mohl zúčastnit vývoje smysluplných projektů, které by měly po sléze i reálné využití. Dále jsem si chtěl vyzkoušet práci v týmu a zapracovat na svých tzv. soft skills, které jsou pro pohodový chod v týmu zapotřebí. Podobná příležitost se mi naskytla ve 4. semestru, kde jsem byl na Erasmu v Belgii a jeden z předmětů byl zaměřen na tvorbu reálného projektu. Pod vedením profesora Wima De Bruyna jsme s hrstkou ajťáků pomáhali malé nemocnici v Madagaskaru navrhnout a implementovat Informační Systém pro správu léků. Ačkoliv byla celá realizace komplikována situací v Madagaskaru, kde často docházelo k výpadkům proudu nebo internetu a tudíž komunikace s nimi byla omezená, tak mi tato forma studia připadala velice atraktivní. Při volbě tématu bakalářské práce jsem měl tedy jasno na čem bych rád pracoval.

Na následujících pár stránkách bych vám rád stručně popsal čemu se společnost ECHOpix věnuje a na jakém poli působí. Dále pak podám bližší informace k jednotlivým úkolům, kterými jsem byl pověřen. Konkrétně tedy vývoj dvou mobilních aplikací. První z nich se jmenuje Oskárek a jednalo se o soutěžní aplikace na zakázku třetí stranu. Druhá nese název Dohodnuto a jedná se o interní aplikaci přímo ECHOpixu, která si klade za cíl jednoduchou tvorbu smluv. Bližší informace k jednotlivým projektům - přesné zadání práce, mé postupy při řešení úkolů, jejich časovou náročnost a využití technologie zmíním níže v podkapitolách věnovaným právě jim.

Na závěr pak zhodnotím svou zkušenost a pokusím se srovnat, jaké znalosti ze školy jsem mohl uplatnit v pracovním prostředí na daných úkolech a jaké předměty tedy pro mě byli na škole stěžejní.

2 Základní informace o firmě

2.1 Odborné zaměření firmy

ECHOpix s.r.o. je mladá, dynamická a neustále se rozvíjející společnost z Krnova, založena v roce 2012. Zabývá se tvorbou reklam, multimediálních projektů, reklamních kampaní, webů, grafikou a jelikož se poptávka po mobilních aplikacích v poslední době neustále zvyšuje, tak začala aktivně působit i na poli vývoje mobilních aplikací. Návrhy aplikací jsou buď interní nebo jsou často zažádány třetí stranou.

2.2 Mé pracovní zařazení

Společně se spolužákem jsme byli přijati do zmíněné firmy a vytvořili dvoučlenný tým. Pod odborným dozorem jsme dostali na starost vývoj dvou aplikací. Práci jsme strukturovali na sérii menších úkolů a ty jsme si pak rozdělili mezi sebe a pečlivě na nich pracovali. Pravidelně jsme se pak scházeli s naším vedoucím a ostatními členy, co nás měli na starosti a prezentovali jim dosavadní postup a vzájemně jsme si kladli dotazy kam a jak směřovat další kroky vývoje.

3 Detailní přehled jednotlivých úkolů

3.1 Společné technologie a nástroje

Díky bakalářské praxi jsem se dostal do kontaktu s mnoha nástroji a technologiemi ulehčující vývoj SW. S některými z nich jsem měl zkušenosti již z práce ve škole a o jiných jsem pouze věděl, že existují a až právě díky praxi jsem si je měl tu příležitost vyzkoušet. Níže zkusím popsat ty, které pro mě po dobu působení ve firmě hrály největší roli.

Jelikož jsou oba projekty primárně zaměřené na mobilní zařízení s operačním systémem Android, využíval jsem především Android framework s IDE Android Studio. U Dohodnuta se pak jedná o rozsáhlý projekt na delší trať, kde je potřeba udělat i stejně rozsáhlou aplikaci pro webové rozhraní a bude následovat i verze pro iOS, takže na vývoji se pak podílím i mimo rámec své bakalářské práce.

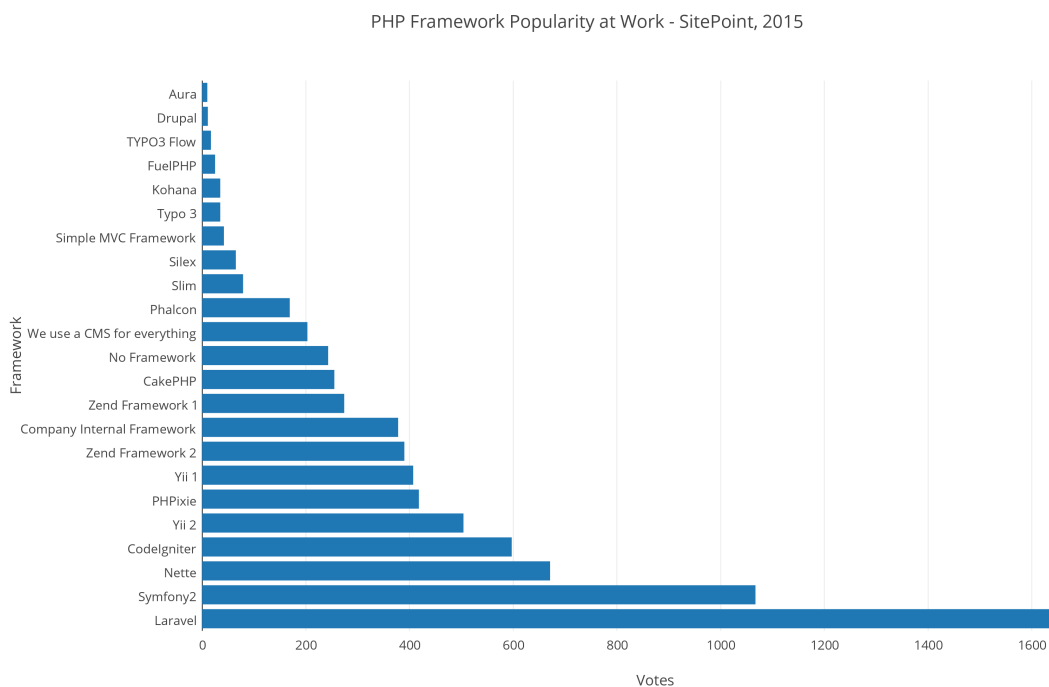
NFC, neboli Near Field Communication. Jedná se o modulární technologii radiové bezdrátové komunikace na krátké vzdálenosti v rámci několika centimetrů (většinou do 4cm). Pomocí této technologie jsme byli schopni rychle přenášet data mezi jednotlivými tablety v soutěžní aplikaci Oskárek.

V Android Studiu jsme pak nejvíce využívali Floobits[7]. Jedná se o plugin, který umožňuje real-time kolaborativní kódování, tedy lidé propojení tímto pluginem vidí, kdo na čem právě pracuje a co přesně píše. Kdykoliv jsme se mohli navzájem přivolat na určitý řádek kódu, prodiskutovat patřičný problém, nebo pracovat na jednom větším souboru současně. Celý projekt je pak uložen v cloudu, tak jsme jeho aktuální verzi měli kdykoliv k dispozici. Tohle bylo velké plus, když jsme museli stihnout důležitou uzávěrku a docházel nám čas, mohli jsme se připojit z jakéhokoliv zařízení a upravovat tak patřičné nedostatky nebo pokračovat v nedokončené práci i doma.

Git je dnes standardním nástrojem pro správu verzí při vývoji jakéhokoliv SW. I my jsme tedy jednotlivé funkční fáze projektu verzovali a zálohovali pomocí Gitu na GitHub. V případě jakéhokoliv závažnějších problémů jsme se tak mohli kdykoliv vrátit k starším a funkčním verzím projektů.

Laravel je open-source PHP framework pro tvorbu webových aplikací založených na architektuře model-view-controller (MVC). Tento framework u nás není tak běžný jako v ostatních zemích světa, ale podle výsledků ankety z roku 2015 na serveru sitepoint.com se jedná o nejpopulárnější PHP framework vůbec, v těsné blízkosti je pak Symfony2 a na třetím místě i Nette, které je nejpoužívanější PHP framework právě u nás v ČR, možná právě díky tomu, že je od českého vývojáře Davida Grudla. Na obrázku níže můžeme vidět, že Nette se umístilo na třetím místě v popularitě s 770 hlasy, ovšem 80% (611) hlasů bylo právě z ČR, v zahraničí se o Nette zase tolik neví. Musíme ale brát v potaz, že tohle jsou výsledky pouze jedné online ankety a reálné čísla se mohou diametrálně lišit. Mezi hlavní výhody Laravelu podle mě patří široká škála připravených balíčků, které vyhovují standardům a mají za sebou několik let správy pod ob-

rovskou komunitou, tudíž eliminujeme chyby (zejména ty bezpečnostní), které by mohly nastat, kdybychom veškeré základní věci měli implementovat od píky sami. Dále pak má skvěle vyřešené ORM, které je kompatibilní s většinou SQL databází a lze implementovat spoustu věcí bez SQL dotazů právě pomocí Laravel syntaxí. O Laravel a webovou část celkově se více staral kolega, který vytvořil celou její kostru a já mu občas s něčím pomohl, když nás tlačil čas, ale primárně jsem měl na starost vývoj mobilní aplikace pro pro Android.

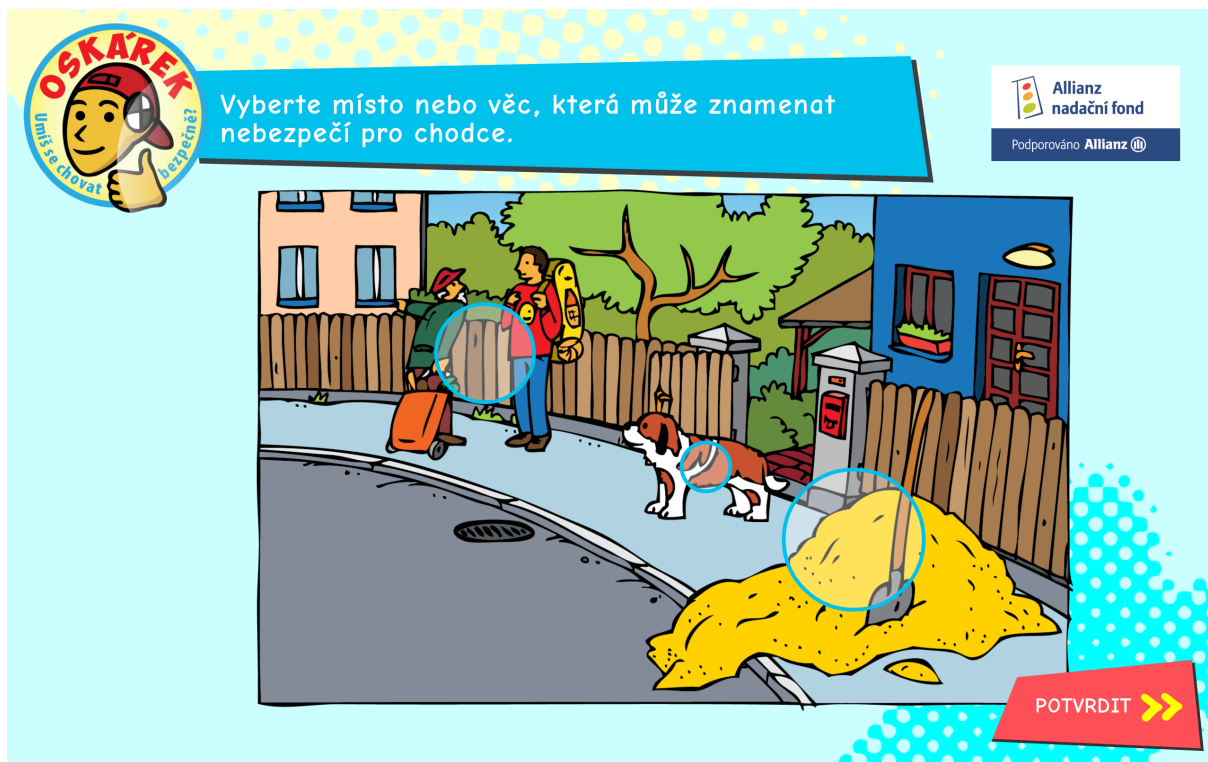


Obrázek 1: Popularita PHP frameworků [11]

3.2 Aplikace Oskárek

3.2.1 Zadání

Jeden ze dvou úkolů, který mi byl na praxi přidělen, byl vývoj soutěžní aplikace Oskárek na zakázku třetí strany, konkrétně od pojišťovny Allianz, která pořádá výukové akce pro děti základních škol. Tato aplikace měla sloužit jako výukový program, pomocí které by se naučili správnému chování na cestách a jiných pozemních komunikacích jak pěšky, tak i na kole nebo jak se správně chovat při jízdě v autě. Základní idea celého projektu byla taková, že budou čtyři stanoviště, mezi kterými budou žáci přebíhat a plnit na nich soutěžní úkoly. Na jednotlivých stanovištích je pak umístěn tablet s aplikací Oskárek, která má za úkol zobrazovat soutěžní otázky korespondující danému ročníku a stanovišti.



Obrázek 2: Ukázka aplikace Oskárek

3.2.2 Technická stránka

Začali jsme tedy analýzou projektu a zpracování požadavků zadavatele pomocí UML diagramů a různých skic. Mezi první úskalí patřilo řešení komunikace zmíněných tabletů. Výukové akce se budou odehrávat na různých lokacích a komunikace by měla být co nejlevnější, nejspolehlivější a nezávislá na okolních podmínkách (signál, WiFi, internet...). Dále následovalo vypracovat návrh vhodných tabletů, které by byly do daných podmínek nejideálnější.

U komunikace byly tři jakési nápady, mezi kterými jsme se po sléze rozhodovali. První byl propojit tablety pomocí WiFi sítě, čili mít centrální router, ke kterému by se všechny tablety připojily a vyměňovaly si informace tímto způsobem. Druhý nápad bylo využít internetového připojení od nějakého operátora. A třetí nápad byl použít NFC technologii a přenášet informace pomocí NFC karet. Jelikož si bylo potřeba posílat jen velice malé množství informací a pouze v určitou dobu, tedy při příchodu soutěžního týmu na stanoviště a potřebovali jsme mít komunikaci nezávislou na lokaci soutěžní akce, zbyla nám více méně jenom jedna možnost a to komunikace formou NFC a programovatelných NFC karet.

Krok číslo dva bylo najít a vybrat vhodné tablety do prostředí, ve kterém se budou soutěžní akce odehrávat. Stěžejní kritéria při jejich výběru byla co největší výdrž baterie, NFC čtečka, která je u většiny tabletů na trhu spíše vzácnost a odolnost, jelikož tablety budou často přenášeny, bude se kolem nich motat spousta lidí a budou situovány ve venkovním prostředí. Našli jsme hrstku tabletů, které těmto požadavkům vyhovovaly a zadavatel se pak na základě našeho

doporučení pro jeden z nich rozhodl. Tablety se objednaly a jakmile jsme je měli po ruce, mohli jsme začít s vývojem.

Při čekání na tablety jsme mohli řešit další věci spojené s vývojem aplikace, tedy zajistit NFC karty nebo štítky, pomocí kterých si budou tablety vyměňovat informace. Na internetu lze sehnat karty různých technologií, například Mifare Ultralight, NTAG203 nebo Mifare 1k. Těch typů je celá řada, ale v podstatě se liší pouze velikostí paměti a přístupem k ní. Jelikož nám o velikost paměti zase tolik nešlo, primárně bylo pro nás důležité mít něco bytelného a většího, co vydrží, neztratí se a na co by se dal nalepit potisk, rozhodli jsme se pro variantu plastových karet Mifare.

3.2.3 Programovací stránka

Po vyřešení technických aspektů projektu jsme se mohli vrhnout na programování samotné. Nejprve bylo potřeba vyřešit čtení a zápis NFC karet. Ze zadání jsme vyvodili, že je potřeba přenášet následující informace:

- Identifikační číslo karty, které je shodné s fyzickým číslem natisknutým na kartě.
- Identifikační číslo týmu, pro zápis do databáze.
- Ročník, abychom věděli, kterou otázku na stanovišti inicializovat.
- Název týmu, abychom ho mohli v aplikaci oslovovat a trochu to tak zlidštit.
- Výsledky jednotlivých otázek ve formě: x/y, kde x je počet získaných bodů a y je maximální počet bodů otázky.

Jednotlivé položky jsou odděleny středníkem, příklad správného formátu dat je pak například "1;2;3;Berušky;1/1;1/1;0/1;2/2", kde se jedná o kartu s identifikačním číslem 1, tým s identifikačním číslem 2, ze 3. ročníku a s názvem Berušky, který splnil všechny 4 soutěžní otázky - první na 1 z 1 bodu, druhou také na 1 z 1 bodu, třetí bohužel na 0 z 1 bodu a poslední čtvrtou na 2 ze 2 bodů.

Jakmile bylo jasné, co přesně na kartu zapisovat, musel jsem zjistit, jakým způsobem ke kartě přistupovat. Naštěstí má Android framework přichystanou sadu metod pro komunikaci s NFC kartami, takže to nakonec bylo relativně snadné. Stěžejní metoda se pak nazývá `onTagDiscovered(Tag tag)`[1], která se při správné implementaci bude starat o detekci karty. Kdykoliv se tedy karta přiblíží k dostatečné blízkosti čtečky, Android vytvoří `Intent`[2] a odešle ho do aplikace, kde se spustí tato metoda. Z tagu přijímaného v parametru si pak vytáhneme `NDEF` objekt[3], pomocí kterého můžeme nad tagem provádět různé operace, jako je čtení a zápis. Data s kterými při těchto operacích pracujeme jsou ve formátu `NdefMessage`[4], což je jakási obálka, která může obsahovat více `NdefRecordů`[5]. `NdefRecord` pak reprezentuje data v binární podobě a může být různého typu (MIME-type media, URI nebo payload aplikace). V našem případě byl plně dostačující základní MIME typ "text/plain".

```
@Override
public void onTagDiscovered(Tag tag){
    Ndef ndef = Ndef.get(tag);
    NdefRecord record = NdefRecord.createMime(MIME_TEXT_PLAIN, strMessage.
        getBytes());
    NdefMessage message = new NdefMessage(record);
    boolean success = false;

    try{
        ndef.connect();
        if(ndef.isConnected()){
            ndef.writeNdefMessage(message);
            success = true;
        }
    }
    catch(IOException | FormatException e){
        Log.i(TAG, e.getMessage());
    }
    finally{
        try{
            if(success){
                ndef.close();
            }
        }
        catch(IOException e){
            Log.i(TAG, e.getMessage());
        }
    }
}
```

Výpis 1: Ukázka kódu zápisu na NFC Tag

Další úskalí leželo v nějakém rozumném zpracování soutěžních otázek. Nechtěli jsme každou otázku programovat zvlášť a jelikož jsme mezi nimi našli při analýze určité podobnosti, byli jsme schopni je rozdělit do 4 typů, občas však bylo nutné některé otázky mírně předělat tak, aby byla zachována pointa a daly se naroubovat na jeden z našich čtyř generických typů. První z nich byl výběr jedné nebo více správných odpovědí. Druhý typ bylo označit jeden nebo více správných elementů na jednom hlavním obrázku (př. "Vyberte místo nebo věc, která může znamenat nebezpečí pro chodce."- na výběr pak byla možnost označit psa, spadlou větev, kopku

písku nebo chodce s velkými taškami - viz. Obrázek 2). Třetí typ byl podobným druhému, s tím rozdílem, že se elementy musely přesouvat na dané místo a měnit i referenci na obrázek. Zkusím to lépe popsat na příkladu přímo soutěžní otázky "Jaké oblečení si zvolíte pro jízdu na kolo?", kde bylo na výběr několik druhů oblečení, které se po označení mělo navléct přímo na postavičku v hlavním obrázku. Například reflexní vesta, umístěna na věšáku u dveří. Kdybychom ji pouze změnili pozici na správné místo, tak by na postavičce nevypadala jako oblečená. Opět jsme tedy využili hashovací tabulky[8], jak pro změnu pozice tak i pro změnu reference na obrázek. Zpracování jsme si pak férově rozdělili na půlky - já jsem naprogramoval a otestoval dva typy a kolega druhé dva typy. Každá otázka pak byla zanesena do hashovací tabulky a při spuštění dané otázky si v ní vyhledala, jakého typu by se měla vytvořit. Díky tomuto způsobu jsme docílili jakési dynamičnosti a přidávat další otázky pak bylo velice snadné.

Další z požadavků klienta byla možnost si otevřít jakoukoliv otázku nezávisle na stanovišti, proto bylo zapotřebí vytvořit jednoduché administrativní rozhraní, ve kterém by měli moderátoři přístup ke všem otázkám a odpovědím, zároveň bylo nutné vypracovat určité zhodnocení soutěžních událostí, jelikož zadavatel chtěl vidět jejich statistiku, jak celkovou v rámci všech událostí, tak konkrétní pro danou událost. K tomu všemu sloužilo zmíněné administrativní rozhraní. Implementace tohoto požadavku byla nenáročná - podobné úlohy jsme řešili ve škole často. Navrhl jsem tedy jednoduchou lokální databázi, příslušné třídy a jejich ORM. O zobrazování a výpis dat se pak staral kolega.

Všechny eventy (2)		Celkové vyhodnocení						
03.03.2016	Krnov	Ročník	1. Otázka	2. Otázka	3. Otázka	4. Otázka	Celkem	Počet lidí
09.03.2016	Testovací	1	100 %	100 %	100 %	100 %	100 %	1
		2	100 %	0 %	0 %	0 %	25 %	1
		3	100 %	100 %	100 %	100 %	100 %	1
							75 %	3
		Týmové vyhodnocení						
Název	Ročník	1. Otázka	2. Otázka	3. Otázka	4. Otázka	Celkem	Počet lidí	
Tým #3	1	100 %	100 %	100 %	100 %	100 %	1	
Tým #4	2	100 %	0 %	0 %	0 %	25 %	1	
Tým #5	3	100 %	100 %	100 %	100 %	100 %	1	
						75 %	3	

ZPĚT

Obrázek 3: Ukázka zhodnocení událostí

3.3 Aplikace Dohodnuto

3.3.1 Zadání

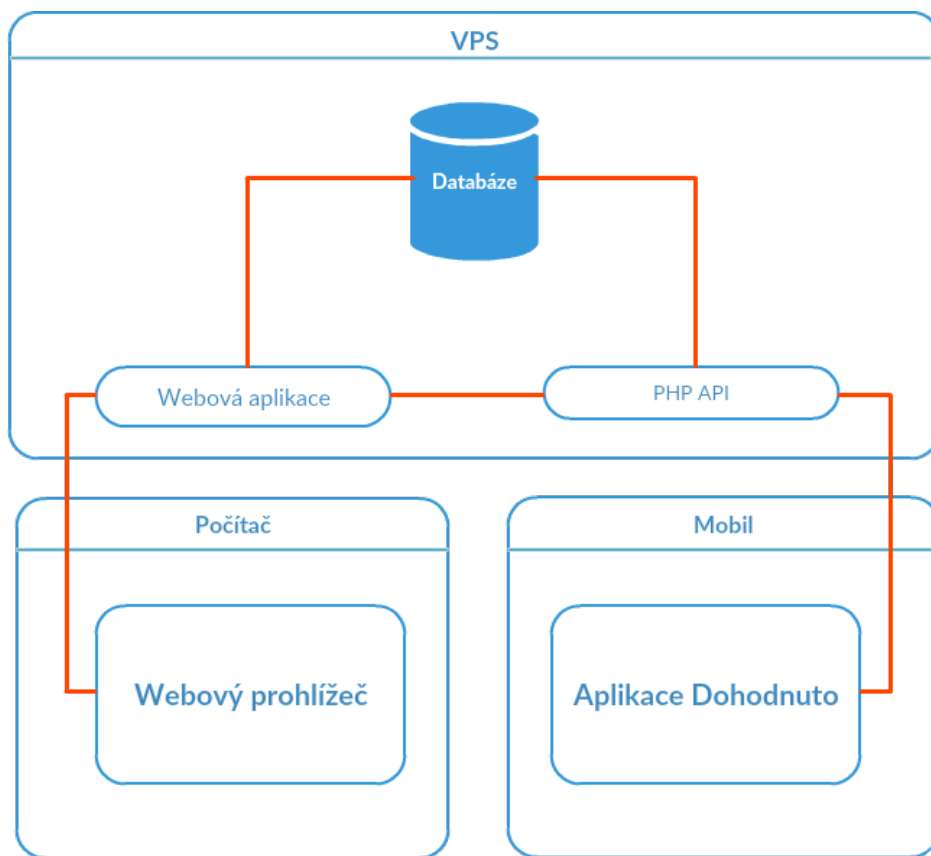
Jedna z interních aplikací firmy je smluvní aplikace Dohodnuto. Jedná se o aplikaci na legální tvorbu smluv podle předem definovaných vzorů, například vzor smlouvy k pronájmu, kupní smlouva nebo smlouva o výpůjčce. V podstatě jde o jednoduché smlouvy, s kterými se v životě setkal snad každý, ale i tak vám dokáží zamotat hlavu a bez pomoci někoho, kdo tomu rozumí, se můžou jevit jako komplikované. Pokud uživateli nevyhovuje ani jeden z nabízených, nebo má vlastní smlouvy, případně by aplikaci rád využíval v rámci své firmy a má firemní vzory smluv, které by rád pomocí této aplikace vyřizoval, může si vytvořit vlastní vzor podle svých představ. Vyřizování smluv se pak stane velice rychlou a snadnou záležitostí, stačí k ní mít u sebe pouze chytrý telefon, kterým v dnešní době disponuje snad každý z nás. Jakmile si uživatel zvolí jakou smlouvu by rád vytvořil, aplikace se ho formou jednoduchých dotazů zeptá na několik potřebných otázek a podle jejich odpovědí pak vygeneruje kompletní a zcela legální smlouvu. Následně si ji lze přečíst, zkontrolovat a v konečné fázi i digitálně podepsat přímo v mobilním zařízení. Smlouva je pak poslána oběma stranám formou PDF a zálohována u nás na server. K smlouvám má tak uživatel kdykoliv přístup, jak z mobilního zařízení, tak z webu.

Hlavním cílem tedy bylo navrhnout a implementovat dvě platformy - mobilní a webovou, ke kterým budou mít zaregistrovaní uživatelé přístup. Obě platformy mají skoro stejnou funkcionalitu, webová část je ještě obohacená a o možnost vytvořit si vlastní smluvní vzor. V mobilní aplikaci tohle nelze, usoudili jsme, že to nemá smysl implementovat, jelikož je daleko pohodlnější a rychlejší tohle řešit na větší obrazovce počítače. Vypisovat a formátovat větší kusy textu na mobilu taky není zrovna příjemné, kvůli zmíněné velikosti obrazovek a softwarové klávesnici. Jestli i přesto bude mít někdo zájem o tvorbu vlastního vzoru na mobilním zařízení, může tak učinit skrz web, který je plně responzivní.

O vývoj android aplikace jsem se převážně staral já, o webovou platformu pak kolega, funkce v API, ke které obě platformy přistupují jsme psali vzájemně podle toho, co kdo zrovna potřeboval.

3.3.2 Technická stránka

Co se týče technické stránky projektu Dohodnuto, museli jsme najít přijatelný hosting, který by zvládl nápor uživatelů z obou aplikací. V raných fázích vývoje jsme koketovali s AWS, konkrétně jejich nabízenou instancí EC2[9]. Jedná se o vysoce kvalitní VPC, u které Amazon navíc nabízí první rok služby zdarma pokud nepřekročíte stanovená pravidla. Tohle se jevilo jako ideální volba v raných fázích vývoje a testování aplikace, díky rychlosti a nulovým nákladům. Poté proběhla plánovaná migrace na vhodnou VPS od firmy OVH, která nabízí kvalitní služby za velice přijatelné ceny a navíc máme možnost podle potřeby jednoduše a rychle upgradovat HW této VPS. Na zmíněné VPS je nainstalovaný Linux s distribucí Debian, na které běží naše PHP API, webová aplikace a MySQL databáze.



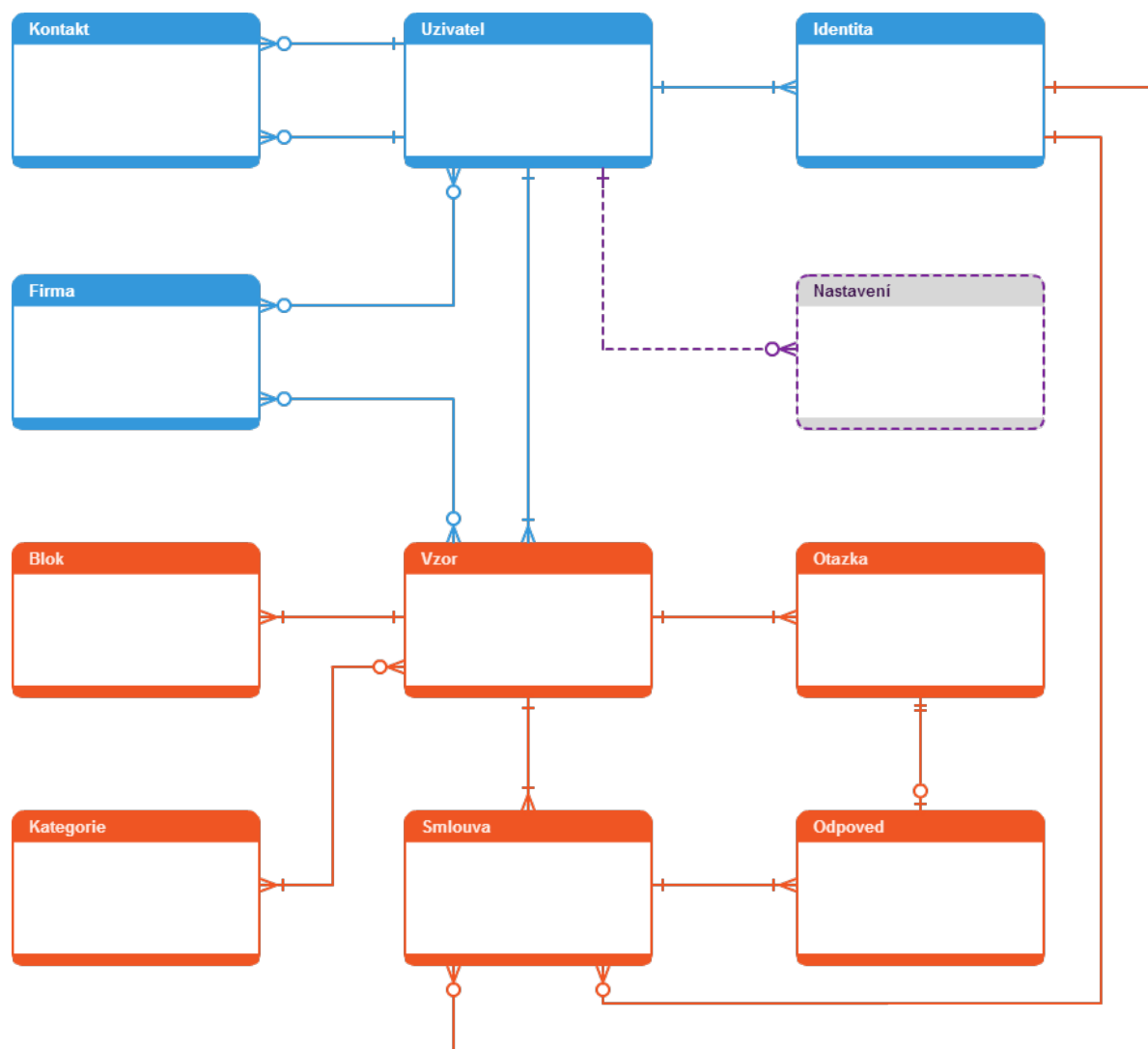
Obrázek 4: Struktura projektu Dohodnuto

Dále bylo potřeba vymyslet jednotnou vrstvu pro komunikaci s databází, která by běžela na zmíněné VPS a předávala data pro obě aplikace. V první řadě vyvstala otázka, zda-li použít REST, nebo SOAP pro komunikaci a po prozkoumání pro a proti těmto možnostem jsme se jednoznačně shodli na RESTu. REST nabízí daleko jednodušší, stručnější, lépe udržitelnou implementaci a pro naše potřeby je plně dostačující. Zůstala však otázka, jestli si tuto službu napsat sami nebo využít různých softwarových balíčků, které by už spoustu věcí řešili za nás. Zkusili jsme krátce využít možnosti číslo dva. Jednalo se o službu zvanou DreamFactory[10], pomocí které jsme mohli sestavit jednoduché API. Jednalo o webovou službu, ve které jste nemuseli nic programovat a vše se dalo "vyklikat" přes jednoduché GUI, tudíž některé věci se velice obtížně implementovaly a celá služba byla značně pomalá. Zjistili jsme, že tudy cesta nevede, jelikož ve výsledku bylo vše složitější, tak jsme se rozhodli, že si celé API napíšeme od puky sami.

3.3.3 Programovací stránka

Hlavní a stěžejní části aplikace co se týče programování pak byl návrh databáze, implementace API, generování smluv a jelikož má aplikace fungovat i bez připojení k internetu tak synchronizace mezi lokální a vzdálenou databází.

Dostali jsme několik smluv zpracovaných právníky tak, aby stačilo vyplnit jen pár slov či vět a byla by správná a kompletní. Podle toho jsme také navrhli databázi, která byla strukturována následovně: hlavní stavební článek pro tvorbu smluv je vzor. Se vzorem jsou pak přímo spjaté otázky a bloky. Otázky jsou rozděleny na několik typů, jako například text, datum, čas, číslo, identity obou stran a jiné. Tyto typy slouží k tomu, abychom v aplikacích mohli zobrazovat správné vstupní políčka, čili když je otázka typu "DATE", zobrazíme v mobilní aplikaci pod otázkou pouze kalendář. Jelikož smlouvy můžou mít dynamickou délku, různé dodatky a podobné věci, závislé na předešlých odpovědích, vymysleli jsme typ zvaný varianta. Ten rozhoduje o tom, jaké další otázky je potřeba zobrazit. Uvedu jednoduchý příklad - u kupní smlouvy patří mezi tento typ otázka "Způsob platby", na kterou lze odpovědět třemi různými způsoby - zapláceno před podpisem, zapláceno bankovním převodem nebo zapláceno na základě daňového dokladu. Podle zvolené možnosti pak zobrazujeme příslušnou pod otázku - jaké je číslo bankovního účtu, na který má platba přijít atd. Podle zvolených variant se také generují bloky, což jsou jednotlivé kusy textu, které při poskládání tvoří kompletní smlouvu. Do každého bloku jsou pak dosazeny odpovědi z otázek. Mimo tuto hlavní část se databáze také skládá z uživatelů, každý uživatel může mít několik identit (právnícká osoba, fyzická osoba podnikatel a fyzická osoba nepodnikatel) a uchovávat si kontakty na jiné uživatele aplikace. Dále může být součástí nějaké firmy, která si, jak už bylo zmíněno na začátku, může vytvořit vlastní firemní vzory pro smlouvy atd.



Obrázek 5: Pseudo struktura databáze

Poté co uživatel odpoví na všechny otázky, vytvoří se instance třídy Smlouva, která je převedena na JSON objekt a poslána do API. Pro komunikaci s API v mobilní aplikaci využívám knihovnu Retrofit, která mi dovoluje vytvořit typově bezpečného HTTP klienta a zároveň se také pomocí knihovny GSON stará o serializaci a deserializaci mezi POJO a JSON. API následně smlouvu zapíše do databáze a pošle zpět zprávu o úspěchu či neúspěchu. Smlouva pak čeká na digitální podpis. Zde jsou na výběr dvě možnosti, podepsat smlouvu na místě v jednom mobilním zařízení nebo odeslat smlouvu k podpisu jinému uživateli aplikace. Poté co ji obě strany podepíší se podpisy posílají do API, která je zpracuje, uloží na server a upozorní uživatele o výsledku této akce pomocí notifikace.

Pro vývoj jednotlivých platform jsme vedli společný seznam s úkoly a věcmi, které je potřeba udělat nebo otestovat. V závislosti na důležitosti nebo očekávání vedení firmy, podle toho, co

by mělo být hotové do určitých termínů se pak může kdokoliv z nás pod určitý úkol podepsat a začít na něm pracovat. Jedná se o takový zjednodušený scrum.

Mobilní aplikace je opět psaná v Javě s Android frameworkem, kde byl kladen důraz na jednoduchý design, který vyhovuje dnešnímu standardu, co se týče rozložení elementů, volby barev a jiných věcí. V době vývoje aplikace se jednalo o tzv. Material design[6]. Ve stejném duchu s důrazem na konzistenci a tvorbu moderního responzivního webu pak probíhala i implementace webová částí, která je psaná v PHP s frameworkem Laravel.

Nová Smlouva
Kupní smlouva - movitá věc

Kdo je prodávající?
Vyberte vlastní identitu, která bude v této smlouvě vystupovat jako prodávající ☐

Kdo je kupující?
Vyberte osobu z adresáře, která bude v této smlouvě vystupovat jako kupující ☐

Movitá věc, nebo materiál?
Vlastníte materiál, či movitou věc? ☐

☐ Movitá věc
☐ Materiál

HOTOVO

Kupní cena
Způsob platby ☐

Obrázek 6: Dohodnuto - Android aplikace

The screenshot shows the 'Dohodnu.to' web application interface. The top navigation bar is red with the logo 'DOHODNU.TO' and links for 'Vzory', 'Smlouvy', 'Adresář', and 'Profil'. The main content area is divided into two columns. The left column contains a sidebar with sections: 'Kupní smlouva - movitá věc' (with a description), 'Vyplnění' (with a pencil icon and 'Vyplnění náležitostí smlouvy'), 'Podepsání' (with a signature icon and 'Podepsání smlouvy odesílatelem'), and 'Odeslání' (with an envelope icon and 'Odeslání smlouvy druhé straně'). Below these is a progress bar showing '0%' and 'Vyplněno 0 z 6 otázek', followed by two buttons: 'Uložit koncept' (red) and 'Dokončit' (green). The right column is a form for creating a purchase agreement. It includes a text input for 'Identita', a dropdown for 'Zvolte uživatele', and several radio button options: 'Movitá věc' and 'Materiál' (selected), 'Zaplacena před podpisem', 'Bude zaplacená na základě daňového dokladu', 'Bude zaplacená bankovním převodem', 'Místem dodání předmětu koupě je sídlo, místo podnikání nebo adresa prodávajícího', 'Místem dodání předmětu koupě je sídlo, místo podnikání nebo adresa kupujícího', 'Místo dodání se liší', 'Předmět koupě byl dnešního dne předán', and 'Předmět koupě bude dodán ve stanoveném datu'. At the bottom of the form is a red 'Odeslat' button.

Obrázek 7: Dohodnuto - Webová aplikace

Při výběru jazyka pro psaní API jsme se rozhodli pro PHP, jelikož je to jednoduchý jazyk k naučení, na webu běží velice svižně, je jednodušší na spravování a hostování než-li jiné jazyky, navíc ve firmě je několik webových vývojářů, kteří nám mohli při vývoji kdykoliv pomoci a po sléze API také spravovat. Využili jsme tedy frameworku zvaného Slim, který si klade za cíl pomoci při psaní jednoduchých, ale výkonných webových API právě v již zmíněném programovacím jazyku PHP. Přístup k API je pak vyřešen tak, že každý uživatel aplikace má přidělen API klíč k jejímu používání, který kontrolujeme před každým požadavkem, aby nedošlo k zneužití nebo přístupu k datům, na které nemá uživatel právo. Zde pak provádíme CRUD a jiné operace, aby byla klientská aplikace co nejméně zatěžována.

Jelikož by měla mobilní aplikace fungovat i bez připojení k internetu, tak mezi jeden z oříšků patřila synchronizace dvou databází - lokální v mobilní aplikaci a té vzdálené na serveru. Každá tabulka tak má speciální sloupec zvaný verze, který se inkrementuje při každé změně. Uživatel může s aplikací pracovat i v offline režimu a po přechodu do online režimu se data aktualizují. Pro zjištění zda-li je nebo není nutná synchronizace kontroluji u každé tabulky počet řádků, součet verzí a nejvyšší hodnotu primárního klíče. Jestli jsou všechny tři hodnoty stejné, mám aktuální verzi databáze, v opačném případě proběhne synchronizace.

4 Závěr

4.1 Časová náročnost projektů

Oba projekty byly poměrně rozsáhlé a zahrnovaly pečlivě provedený návrh, jeho implementaci, nastudování různých technologií a frameworků, práce s databází, od zanesení dat, synchronizace mezi dvěma databázemi až po ORM. V neposlední řadě také testování funkčnosti aplikací a oprava případných chybiček. Díky Floobitům a Gitu jsem mohl v případě nouze dodělat nebo opravit i nějaké věci z domu. K tomu docházelo především v případě, že se pracovalo na externí aplikaci na zakázku a bylo potřeba ji stihnout dokončit ve stanoveném termínu. Celkový časový rozsah, mé bakalářské práce pak vyšel na 25 dní v zimním semestru, 30 dní v letním semestru a 20 dní mimo semestr. Celkový počet odpracovaných dní je tedy 75. Z toho zabralo bezkonkurenčně nejvíc času vývoj interní aplikace Dohodnuto - něco kolem 40-50 dní, zbytek spolkla externí aplikace Oskárek a nepatrné množství i jiné menší projekty.

4.2 Dosavadní znalosti

Díky studiu na škole jsem se seznámil s mnoha programovacími jazyky, které jsem pak hojně využíval i při absolvování práce. Pro práci ve firmě jsem zejména využil Javu, SQL a jejich rozšíření. Dále mi pak přišly vhod předměty, které se zabývaly správnou implementací a stavbou aplikací, kde se probírali osvědčená řešení různých situací a návrhové vzory. V neposlední řadě také předměty, které mě naučili efektivnějším způsobům programování a řešení různých algoritmů. Konkrétně se jednalo o následující předměty:

- Programovací jazyky I, II
- Algoritmy I, II
- Úvod do databázových systémů
- Databázové a informační systémy
- Tvorba mobilních aplikací II
- Úvod do softwarového inženýrství
- Vývoj informačních systémů
- Seminář z programování

4.3 Nabyté znalosti

Všechny teoretické znalosti nabyté ve škole jsem měl konečně možnost uplatnit v reálných projektech a dostat zpětnou vazbu od lidí, kteří se na tomto poli pohybují pravidelně a déle než já.

Mezi další velké pozitivum patřila práce v týmu, kde záleželo na rozdělení úkolů v rámci týmu, precizní vypracování daných úkolů podle zadání, dokončování práce ve stanovených termínech, zálohování a verzování pomocí Gitu a jiné. Dále také střet se zákazníky, kde bylo potřeba dojít k efektivnímu řešení při realizaci návrhu projektů tak, aby byla zajištěna jejich spokojenost. V neposlední řadě také možnost pracovat s novými technologiemi, různými frameworky a špičkovou elektronikou. V rámci studia na škole bych neměl možnost se s tím vším potkat.

4.4 Celkové zhodnocení

Možnost vykonat bakalářskou práci formou praxe pro firmu ECHOpix s.r.o. mi bylo neskutečným přínosem. Měl jsem možnost ověřit si své dosavadní znalosti, které byly z větší míry pouze teoretické a tudíž je konečně využít v praxi. Práce v reálném prostředí a vývoj v týmu mě také nesmírně obohatil a otevřel oči tomu, jak se věci doopravdy dělají. Bylo mi umožněno spolupracovat na zajímavých projektech a využít moderních technologií a mít přístup k elektronice, ke které bych se jinak nedostal. Navíc vznikla dlouhodobá spolupráce s firmou a možností pracovat pro ně i do budoucna. Za mě to byla jedna velká, pozitivní zkušenost a tuto formu bakalářské práce bych doporučil každému, jelikož jsem tím tak zabil ne dvě, ale hned několik much jednou ranou - měl jsem zajištěnou bakalářku, získal praxi a reference do budoucna a byl jsem finančně ohodnocen.

Literatura

- [1] Tag. Android Developers. [online]. [cit. 2016-04-12]. Dostupné z: <http://developer.android.com/reference/android/nfc/Tag.html>
- [2] Intent. Android Developers. [online]. [cit. 2016-04-12]. Dostupné z: <http://developer.android.com/reference/android/content/Intent.html>
- [3] Ndef. Android Developers. [online]. [cit. 2016-04-12]. Dostupné z: <http://developer.android.com/reference/android/nfc/tech/Ndef.html>
- [4] NdefMessage. Android Developers. [online]. [cit. 2016-04-12]. Dostupné z: <http://developer.android.com/reference/android/nfc/NdefMessage.html>
- [5] NdefRecord. Android Developers. [online]. [cit. 2016-04-12]. Dostupné z: <http://developer.android.com/reference/android/nfc/NdefRecord.html>
- [6] Material Design. Google. [online]. [cit. 2016-04-12]. Dostupné z: <https://www.google.com/design/spec/material-design>
- [7] Floobits. Floobits. [online]. [cit. 2016-04-12]. Dostupné z: <https://floobits.com/>
- [8] Hashovací tabulka. Algoritmy. [online]. [cit. 2016-04-12]. Dostupné z: <https://www.algoritmy.net/article/32077/Hashovaci-tabulka>
- [9] Amazon Virtual Private Cloud (VPC). Amazon Web Services. [online]. [cit. 2016-04-12]. Dostupné z: <https://aws.amazon.com/vpc/>
- [10] API automation. DreamFactory. [online]. [cit. 2016-04-12]. Dostupné z: <https://www.dreamfactory.com/>
- [11] The Best PHP Framework for 2015: SitePoint Survey Results. Sitepoint. [online]. 30.3.2015 [cit. 2016-04-24]. Dostupné z: http://dab1nmslvvntp.cloudfront.net/wp-content/uploads/2015/03/1427547421php_framework_popularity_at_work_-_sitepoint2c_2015.png